POLYTECH SORBONNE

Sorbonne Université
2020-2021

Cours de Programmation en C – EI-2I

Travaux Pratiques

Objectif(s)

- ★ Création de fonctions
- ★ Manipulation de tableaux

1 Un peu d'Histoire...[1]

Le cylindre de Jefferson est un système de chiffrement polyalphabétique inventé par Thomas Jefferson (futur président des États-Unis) vers 1793 alors qu'il était secrétaire d'État de George Washington.

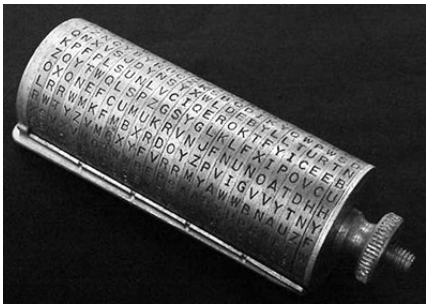


Figure 1: Cylindre de Jefferson[2]

Le principe de fonctionnement du cylindre de Jefferson (*Jefferson's wheel cipher* en anglais) est relativement simple et ingénieux. Il se compose d'un axe autour duquel tourne un certain nombre de roues (Figure 1). Sur la tranche de chacune d'elles, vingt-six lettres, représentant l'ensemble des lettres de l'alphabet latin, sont gravées de façon aléatoire. En tournant les roues les unes par rapport aux autres, il est alors possible d'aligner une suite de caractères de manière à composer un message. Par exemple, un expéditeur désirant écrire le message suivant "Thomas Jefferson wheel cipher", composera sur une même ligne du cylindre, l'ensemble des caractères du message. Pour coder ce dernier, l'expéditeur envoie la succession de caractères d'une ligne située au dessus ou en dessous de la ligne du message. Le destinataire recevra , par exemple, le message suivant "MZNCSKYONSLKTRFAJQQBRTXYUKA" qu'il composera sur son cylindre identique à celui de son correspondant. En recherchant une ligne pour laquelle la succession de caractères est intelligible, le destinataire découvrira le message envoyé.

2 But du TP

Le but du TP est d'écrire un programme afin de recréer le principe de fonctionnement du cylindre de Jefferson. Ce dernier sera assimilé à un tableau 2D de 26 lignes et de N colonnes correspondant au nombre de roues du cylindre (cf figure 2). Dans la suite du TP, le nombre de colonnes N du tableau sera constant et défini avec un `#define`. On prendra $N = 19$.

j	t	s	e	c	b
f	a	h	s	q	g
⋮	⋮	⋮	⋮	⋮	⋮
⋮	⋮	⋮	⋮	⋮	⋮
y	c	b	t	z	c

Figure 2: Tableau représentation le cylindre de Jefferson

Pour chaque nouvelle question, vous devrez fournir un nouveau fichier (par exemple `Jefferson-Q1.c` pour la 1ère question). Reprenez donc le code correspondant à la question précédente (dans un nouveau fichier) pour démarrer une nouvelle question.

[1] Source wikipedia : `http://en.wikipedia.org/wiki/Jefferson_disk`
[2] Source : National Cryptographic Museum Foundation

3 Programmation du cylindre de Jefferson

Question 1 – construction du cylindre – 6pts

Dans le répertoire , nous fournissons un fichier `jefferson.txt` contenant les caractères d'un cylindre de Jefferson (pour 19 roues). Le fichier contient les 26 caractères de la 1ère roue, suivi des 26 de la 2ème roue, puis de la 3ème, etc.
Écrivez deux fonctions `LitCylindre` et `AfficheCylindre` telles que :

- `LitCylindre` permet de lire les caractères contenus dans le fichier et les stocke dans le tableau. Cette fonction prendra comme paramètres :
  - le descripteur de fichier (`FILE*`);
  - le tableau à remplir.

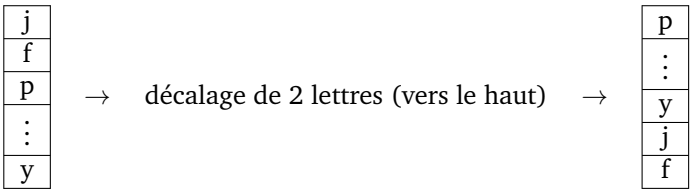
  La fonction renverra un entier indiquant si la lecture s'est bien déroulée.
- `AfficheCylindre` permet d'afficher le cylindre construit (affichage ligne par ligne).

Écrivez une fonction principale `main` qui permet de tester ces fonctions en lisant ouvrant le fichier "`jefferson.txt`" et affichant le cylindre lu.

Question 2 – Manipulation du cylindre – 6pts

Écrivez deux fonctions `chercheLettreRoue` et `tourneRoue` telles que :

- `chercheLettreRoue` permet de déterminer dans une roue du cylindre la position (entre 0 et 25) d'une lettre cherchée (chaque lettre de l'alphabet apparait une fois par roue). Cette fonction prendra comme paramètres :
  - le tableau représentant le cylindre ;
  - le numéro de la roue ;
  - la lettre recherchée.
- `tourneRoue` permet de tourner une roue (vers le haut) d'un nombre précis de lettres. Avant de tourner la roue, et donc de décaler les lettres du bas vers le haut, le plus facile est d'utiliser un tableau intermédiaire pour stocker les lettres afin de ne pas perdre leur ordre d'origine.



Complétez/modifier la fonction principale `main` en choisissant une roue et une lettre à rechercher et placez-la sur la première ligne du cylindre en utilisant la dernière fonction.

Question 3 – Application au chiffrement – 4pts

Constatez que le programme crée peut permettre de coder ET de décoder un message donné en paramètre d'entré. Pour cela, dans un fichier `jeffersonQ3.c` (qui reprend `jeffersonQ2.c`), complétez la fonction `main` pour demander à l'utilisateur d'entrer *N* caractères, soit 19 caractères, (en minuscule), et effectuer les rotations nécessaires des roues afin de placer les caractères à coder sur la première ligne du cylindre.
Lorsque cette opération sera faite, vous pourrez alors décoder le message suivant : "mkcrvylhcoiahcyzkdk".

Question 4 – Ordre des roues – 2pts

En plus de cela, il est intéressant (et important) de pouvoir déterminer l'ordre des roues du cylindre car cela constitue la clé du chiffrage.
Proposez une solution où l'utilisateur doit saisir l'ordre des *N* roues du cylindre, puis un code à chiffrer/déchiffrer.

Question 5 – Bonus – 2pts

Ré-écrivez la fonction `tourneRoue` **sans** utiliser de tableau intermédiaire.